Jar's 1(1) (Desember 2022) 1-11 P-ISSN : 2985-3079

Jurnal Advance Research Informatika

Informatika dan Sistem Informasi

https://www.ejournalwiraraja.com/index.php/FT

APLIKASI ENKRIPSI-DEKRIPSI FILE TEXT MENGGUNAKAN ALGORITMA TINY ENCRYPTION ALGORITHM (TEA)

DONI KURNIAWAN[1,*)], IRWAN DARMAWAN[2)]
Jurusan Teknik Infomatika, Universitas Madura, Pamekasan, Indonesia
EMAIL: doni@gmail.com, darmawan@unira.ac.id

Diterima : 01 November 2022. Disetujui : 05 Desember 2022. Dipublikasikan : 26 Desember 2022.

ABSTRACT *- The security and confidentiality of thesis data is very important, especially if the thesis data is stored in digital form and distributed via digital communication channels, so techniques/ways are needed to secure the data, the methods and techniques commonly used are data randomization or cryptography. In cryptography, many methods are offered to encode data, one of which is the Tiny Encryption Algorithm (TEA). By using the TEA Algorithm which is a symmetric cryptographic algorithm, the strength of this algorithm lies in the Feistel network (including substitution, permutation and modular arithmetic operations) and delta numbers derived from golden numbers. From the trials conducted by the author, the Tiny Encryption Algorithm (TEA) relies on optimal processing speed with maximum results.*

Keywords : *Cryptography, Tiny Encryption Algorithm (TEA), Encryption, Decryption, Thesis*

ABSTRAK *- Keamanan dan kerahasiaan data skripsi sangatlah penting terlebih jika data skripsi disimpan dalam bentuk digital dan didistribusikan melalui jalur komunikasi digital, sehingga dibutuhkan teknik/cara untuk mengamankan data tersebut, metode dan teknik yang umum digunakan adalah dengan mengacak data atau kriptografi. Dalam ilmu kriptografi banyak metode yang ditawarkan untuk menyandikan data salah satunya adalah Tiny Encryption Algorithm (TEA). Dengan menggunakan Algoritma TEA yang merupakan algoritma kriptografi simetri, kekuatan algoritma ini terletak pada jaringan feistel (meliputi operasi substitusi, permutasi dan modular aritmatic) dan bilangan delta yang berasal dari golden number. Dari uji coba yang dilakukan penulis, Tiny Encryption Algorithm (TEA) mengandalkan kecepatan proses yang optimal dengan hasil yang maksimal.*

Kata kunci : *Kriptografi, Tiny Encryption Algorithm (TEA), Enkripsi, Dekripsi, Skripsi*

I. PENDAHULUAN

Teknologi informasi saat ini semakin populer digunakan dalam seluruh aspek kehidupan. Hampir seluruh informasi kini dikelola dalam bentuk digital. Hal ini didukung oleh berbagai keuntungan yang dapat diperoleh seperti kemudahan dalam penyimpanan dan kecepatan dalam pendistribusian. Akan tetapi, dalam menjaga kerahasian data menjadi hal yang sangat penting baik dalam suatu organisasi ataupun milik pribadi.

Keamanan dan kerahasian sangat dibutuhkan dalam dunia komunikasi khusunya dalam dunia maya, dan diperlukan metode khusus untuk meningkatkan keamanan data tersebut agar tetap terjaga kerahasiaannya.

Laporan Tugas Akhir sebagai syarat untuk menyelesaikan kuliahnya untuk mendapatkan gelar sarjana, merupakan project besar dan penelitian yang berharga sehingga perlu dijaga kerahasiaannya, atau paling tidak dapat menghargai hak cipta atas jerih payah yang telah dilakukan mahasiswa

Fakultas Teknik merupakan salah satu Jurusan yang berada di kampus Universitas Madura yang terletak di Kabupaten Pamekasan, yang masih menggunakan metode manual dalam mengelola Laporan Tugas Akhir (TA) yang dibuat oleh mahasiswa terlebih lagi dalam hal pengamanan masih jauh dari kata sempurna

Oleh Karena itu untuk menghindari pembajakan, serta menghargai hak cipta dalam pengamanan data file skripsi diperlukan sebuah Aplikasi enkripsi dekripsi yang dapat membatasi pembajakan secara ilegal dalam mengakses keseluruhan data, tanpa adanya konfirmasi kepada pihak yang bersangkutan.

Berdasarkan latar belakang diatas, penulis membuat aplikasi berbasis *desktop* yang disusun dalam laporan penelitian ini dengan judul "Aplikasi Enkripsi-Dekripsi File Text Menggunakan Algoritma *Tiny Encryption Algorithm* (TEA)" Agar dapat digunakan dalam memproteksi data file Skripsi serta menghargai terhadap hak cipta.

1.1 Perumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan permasalahan sebagai berikut:

a. Bagaimana merancang dan membuat Aplikasi Enkripsi-Dekripsi file skripsi menggunakan Algoritma *Tiny Encryption Algorithm* (TEA) di Fakultas Teknik Universitas Madura Pamekasan?
b. Apakah aplikasi yang akan dibangun dapat meminimalisir tindakan plagiasi terhadap hak cipta Skripsi di Fakultas Teknik Universitas Madura Pamekasan?

1.2 Batasan Masalah

Adapun pembahasan pada tugas akhir ini tidak keluar dari batasan yang sudah ditentukan maka penulis membatasinya pada beberapa batasan berikut :

a. Dalam penelitian ini hanya membahas mengenai Proses Enkripsi dan Dekripsi file data skripsi menggunakan Algoritma *Tiny Encryption Algorithm* (TEA) di Fakultas Teknik Universitas Madura Pamekasan.
b. Proses Enkripsi dan Dekripsi hanya dilakukan pada file data skripsi untuk memproteksi data yang disimpan didalam storage, bukan pada data yang dikirim (ditransmisikan) dalam saluran komunikasi
c. Dalam penelitian ini hanya membahas mengenai proses enkripsi-dekripsi yang dilakukan pada dokumen dalam format *.doc, *.docx, *.pdf, dan *.txt
d. Aplikasi yang dibuat menggunakan bahasa pemrograman PHP

1.3 Tujuan Penelitian

Adapun tujuan dari tugas akhir ini adalah merancang dan membuat aplikasi keamanan data file skripsi di Fakultas Teknik Universitas Madura Pamekasan yang dapat digunakan dalam hal pengamanan data dan menghargai hak cipta agar tidak dapat diganggu ataupun diakses oleh pihak yang tidak berhak meskipun digunakan pada jaringan yang tidak aman, sehingga keamanan data tetap terjaga

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian yang dilakukan adalah untuk mengetahui sejauh manakah keamanan data file skripsi dapat terjaga dengan menggunakan algoritma *Tiny Encryption Algorithm* (TEA) khususnya dalam pengamanan data skripsi di Fakultas Teknik Universitas Madura Pamekasan, serta dengan adanya software yang dirancang nantinya diharapkan akan timbulnya kesadaran menghargai hak cipta seseorang sehingga bagi siapa saja yang ingin melindungi datanya/memproteksi agar lebih terjaga kerahasiaannya.

II. TEORI DASAR

2.1 *Tiny Encryption Algorithm* (TEA)

Tiny Encription Algorithm (TEA) merupakan suatu algoritma sandi yang diciptakan oleh David Wheeler dan Roger Needham dari Computer Laboratory, Cambridge University, England pada bulan November 1994. Algoritma ini merupakan algoritma penyandian *block cipher* yang dirancang untuk penggunaan memory yang seminimal mungkin dengan kecepatan proses yang maksimal.

Sistem penyandian Tiny Encryption Algorithm (TEA) menggunakan proses feistel network dengan menambahkan fungsi matematik berupa penambahan dan pengurangan sebagai operator pembalik selain XOR. Hal ini dimaksudkan untuk menciptakan sifat non-linearitas. Pergeseran dua arah (ke kiri dan ke kanan) menyebabkan semua bit kunci dan data bercampur secara berulang ulang. Tiny Encryption Algorithm (TEA) memproses 64-bit input sekali waktu dan menghasilkan 64-bit output. Tiny Encryption Algorithm (TEA) menyimpan 64-bit input kedalam L0 dan R0 masing masing 32-bit. Sedangkan 128-bit kunci disimpan kedalam k[0], k[1], k[2], dan k[3] yang masing masing berisi 32-bit. Diharapkan teknik ini cukup dapat mencegah penggunaan teknik exshautive search secara efektif. Hasil outputnya akan disimpan dalam L32 dan R32. (D. Wheeler and R. Needham, 1994)

Bilangan delta konstan yang digunakan adalah 9E3779B9, dimana bilangan delta berasal dari golden number $(\sqrt{5}-1)2^{31}$. Berbeda dengan sruktur feistel yang semula hanya mengoperasikan satu sisi yaitu sisi sebelah kanan dengan sebuah fungsi F, pada algoritma Tiny Encryption Algorithm (TEA) kedua sisi dioperasikan dengan sebuah fungsi yang sama TEA merupakan algortima enkripsi blok, dengan blok-blok sebagai berikut:

64-bit input L0 dan R0 @32-bit.
128-bit kunci k[0], k[1], k[2], k[3] @32-bit.

Rumus Enkripsi:
SUM = n * Delta
$L_n = L_{(n-1)}+((R_{(n-1)}$ Shl 4)+K[0]) XOR ($R_{(n-1)}$+SUM) XOR (($R_{(n-1)}$ Shr 5)+K[1]))
$R_n = R_{(n-1)}$ + ((L_nShl 4)+K[2]) XOR (L_n+SUM) XOR ((L_n Shr 5)+K[3])

Rumus Dekripsi:
SUM = SUM - Delta (SUM = Delta X 32)
$R_n = R_{(n-1)}$ – ((L(n-1) Shl 4)+k[2]) XOR (L(n-1)+SUM)
XOR ((L(n-1) Shr 5)+k[3])
$Ln = L_{(n-1)}$ – ((R_n Shl 4)+k[0]) XOR (R_n + SUM)
XOR ((R_n Shr 5)+k[1]))

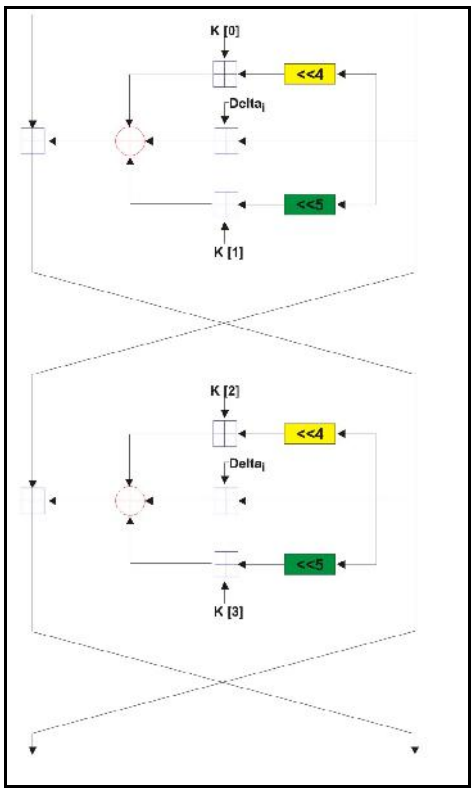
Keterangan:
n = Round ke
Delta = $(\sqrt{5}-1)2^{31}$ = $9E3779B9_{16}$ atau $10011110\ 00110111\ 01111001\ 10111001_2$
L = ciper Left (Kiri)
R = ciper Right (Kanan)
L/R shl4 = Pergeseran bit kekiri sebanyak 4 bit
L/R shr5 = Pergeseran bit kekanan sebanyak 5 bit
K[0,1,2,3] = Key Schedule (Kunci)

Untuk melakukan enkripsi, proses diawali dengan input-bit file biner sebanyak 64-bit. Kemudian 64-bit biner tersebut dibagi menjadi dua bagian, yaitu sisi kiri (Left) sebanyak 32-bit dan sisi kanan (Right) sebanyak 32-bit. Setiap bagian akan dioperasikan sendiri-sendiri. R(n-1) akan digeser kekiri sebanyak empat (4) bit dan ditambahkan dengan kunci k[0]. Sementara itu R(n-1) ditambah dengan sum (delta) yang merupakan konstanta. Hasil penambahan ini di-XOR-kan dengan penambahan sebelumnya. Kemudian di-XOR-kan dengan hasil penambahan antara R(n-1) yang digeser kekanan sebanyak lima (5) bit dengan kunci k[1]. Hasil tersebut kemudian ditambahkan dengan L(n-1)yang akan menjadi Ln. Sisi sebelah kanan akan mengalami proses yang sama dengan sisi sebelah kiri. (Ln) akan digeser kekiri sebanyak empat (4) bit lalu ditambahkan dengan kunci k[2]. Sementara itu, (Ln) ditambah dengan sum (delta). Hasil penambahan ini di-XOR- kan dengan penambahan sebelumnya. Kemudian di-XOR-kan dengan hasil penambahan antara (Ln) yang digeser ke kanan sebanyak lima (5) bit ditambahkan Kunci k[3]. Hasil tersebut kemudian ditambahkan kembali dengan R(n-1) yang akan menjadi L1. Secara matematis dapat ditulis seperti berikut :

LI = L0 + ((R0 shl4)+K(0)) XOR (R0+SUM) XOR ((R0Shr5)+K(1))

R1 = R0+((L1 Shl4)+K(2) XOR (L1+SUM) XOR ((L1Shr5)+K(3))

Berikut adalah gambaran proses pada algoritm a TEA:



Gambar 1.
Algoritma Tea

Berikut adalah langkah langkah penyandian dengan algoritma TEA dalam satu cycle (dua round) :

1. Pergeseran (shift)

Inputan file biner pada kedua sisi yang masing masing sebanyak 32-bit akan digeser kekiri (*Left Shift*) sebanyak empat (4) bit dan digeser ke kanan (*Right Shift*) sebanyak lima (5) bit.

2. Penambahan

Setelah mengalami pergeseran (shift) bit, maka L dan R yang telah digeser akan ditambahkan dengan kunci 128 bit yang disimpan ke dalam K[0], K[1], K[2], K[3] yang masing-masing berisi 32 bit. Sedangkan L dan R awal akan ditambahkan dengan sum (delta).

3. Operasi XOR

Setelah proses pergeseran dan penambahan pada masing-masing register maka akan dilakukan operasi XOR, dimana dalam setiap bagian terdapat dua kali operasi XOR. dengan rumus untuk satu roundnya adalah sebagai berikut :

L = L + (((R<<4)+K[0]) **XOR** R+sum **XOR** ((R>>5)+k[1]))
R = R + (((L<<4)+k[2]) **XOR** L+sum **XOR** ((y>>5)+k[3]))
(dalam hal ini sum=sum+delta.)

4. Key Schedule (Kata Kunci)

Kata kunci Pada algoritma TEA, terdiri dari 128 bit kunci yang disimpan kedalam k[0], k[1], k[2], dan k[3], untuk penggunaannya kunci k[0] dan k[1] konstan digunakan untuk round ganjil sedangkan kunci k[2] dan k[3] konstan digunakan untuk round genap. Apabila Input kata kunci melebihi batas, maka akan dihapus secara otomatis. Dalam proses dekripsi sama halnya seperti pada proses penyandian yang berbasis feistel cipher lainnya. Yaitu pada prinsipnya adalah sama pada saat proses enkripsi. Namun hal yang berbeda adalah penggunaan teks sandi sebagai input dan kunci yang digunakan urutannya dibalik. Pada proses dekripsi semua round ganjil menggunakan k[1] terlebih dahulu kemudian k[0], demikian juga dengan semua round genap digunakan k[3] terlebih dahulu kemudian k[2].

Adapun beberapa keunggulan dari algoritma Tiny Encryption Algorithm (TEA) ini adalah :

a. Pada Algoritma Tiny Encryption Algorithm (TEA) panjang kuncinya yaitu 128-bit, merupakan jumlah kunci yang cukup panjang untuk algoritma kriptografi modern saat ini yang dapat menahan serangan kriptanalis.
b. Teknik yang digunakan TEA cukup baik, yaitu pada setiap prosesnya menggunakan jaringan feistel yang memuat operasi permutasi, subtitusi dan modular arithmatic berupa XOR dan penambahan bilangan delta yang diharapkan dari operasi tersebut menciptakan efek difusi dan konfusi yang baik, karena semakin baik efek difusi dan konfusi yang dihasilkan suatu algoritma makin semakin baik pula tingkat keamanannya.

c. Ukuran blok input pada TEA yaitu 64-bit, sebuah jumlah yang cukup panjang untuk menghindari analisis pemecahan kode dan cukup kecil agar dapat bekerja dengan cepat.
d. Tidak membutuhkan S-Box dan P-Box dalam proses enkripsi dan deskripsinya, karena S-Box dan P-Box tersebut tidak dapat dijamin keamanannya dikarenakan struktur dari S-Box dan P-Box tersebut hanya diketahui oleh NSA (National Security Agency) dan diubah menurut saran dari NSA, sehingga jika S-Box dan P- Box tersebut diubah maka sangat mungkin sekali algoritma yang digunakan akan lebih mudah dibobol. Selain itu, juga dapat meminimalkan penggunaan memory pada saat melakukan proses enkripsi dan deskripsi sehingga dapat memaksimalkan proses.
e. Algoritma TEA diketahui sangat kuat terhadap metode penyerangan berupa hanya ciphertext yang diketahui, plaintext yang diketahui dan plaintext terpilih. Sedangkan kelemahan dari algoritma Tiny Encryption Algorithm (TEA) ini adalah karena TEA ini termasuk kedalam kelompok Algoritma Simetri, maka masih rentan untuk dibobol, karena dalam algortima simetri masalah utama memang terletak dari segi pendistribusian kuncinya, dimana harus benar-benar aman pada saat mendistribusikan kunci yang akan digunakan. Berdasarkan data yang didapat, estimasi proses enkripsi dan deskripsi algoritma TEA yang dibandingkan dengan algoritma simetri lainnya secara umum adalah sebagai berikut :

Tabel 1.
Perbandingan Estimasi Proses Algoritma TEA deng an Algoritma Simetri lainnya

Type	**Author**	**Block Size Bits**	**Key Size Bits**	**Speed (m:s)**
3DES	Diffie & Hel lman	64	168	4:05
Blowfish	Schneier	64	256	0:55
DES	IBM & NSA	64	56	1:42
IDEA	Lai & Masse y	64	128	1:07
Misty1	Matsui	64	128	2:50
Square	Daemon & Rijmen	128	128	0:39
Summer	Aman	(strea m)	128	0:46
TEA 16	Wheeler & Needham	64	128	0:46
TEA 32	Wheeler & Needham	64	128	1:03

III. RANCANGAN SISTEM

Tahapan analisis terhadap suatu sistem dilakukan sebelum tahapan perancangan dilakukan. Adapun tujuan dilakukannya analisis terhadap suatu sistem adalah untuk mengetahui alasan mengapa sistem tersebut diperlukan, yaitu dengan merumuskan kebutuhan-kebutuhan dari sistem tersebut untuk meminimalisir sumber daya yang berlebih serta membantu merencanakan penjadwalan pembentukan sistem, meminimalkan distorsi-distorsi yang mungkin terdapat di dalam sistem tersebut sehingga dapat bekerja secara optimal.

Sistem ini membantu untuk pengamanan data dan sebagai media untuk menghargai hak cipta agar tidak dapat diganggu maupun diakses oleh pihak yang tidak berhak, dokumen skripsi yang dimasukkan kedalam database di enkripsi dengan menggunakan kunci oleh admin, sehingga apabila ada yang menginginkan data tersebut dapat di dekripsi menggunakan kunci yang sama dengan menghubungi admin fakultas. Adapun proses Enkripsi dan Dekripsinya menggunakan metode *Tiny Encryption Algorithm (TEA).*

3.1 Arsitektur Sistem

Pengguna yang menginginkan keamanan pada file data yang dimiliki menginputkan file teks pada aplikasi *Data Encryptor* dan melakukan proses enkripsi dengan menginputkan kata kunci. File yang sudah di enkripsi tersebut akan membentuk file baru yang sebelumnya sudah ditentukan letaknya. Untuk dapat mengembalikan file sama seperti semula maka pengguna harus melakukan proses deskripsi,

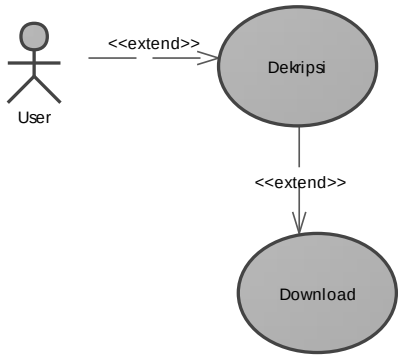
dengan memasukkan kunci yang sama seperti proses Enkripsinya.

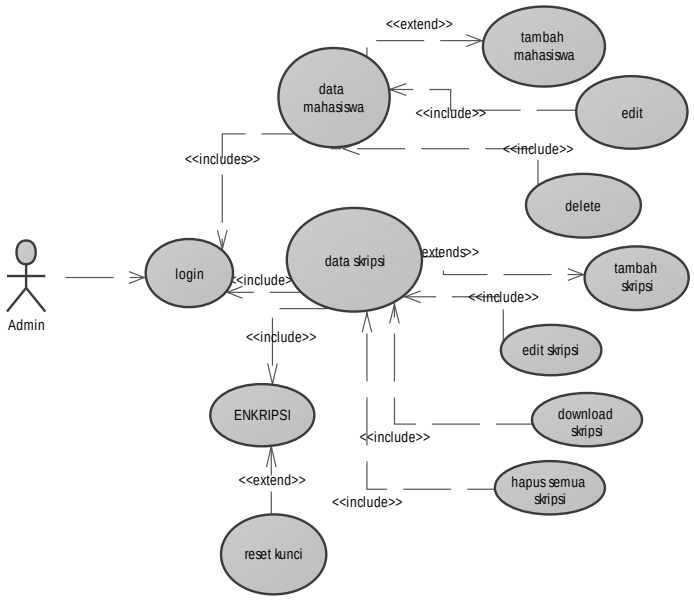
Gambar 2.
Arsitektur Umum Sistem

3.2 Usecase Diagram

Usecase diagram merupakan alat komunikasi tingkat tinggi untuk mewakili persyaratan sistem. Diagram menunjukkan interaksi antara pengguna dan entitas eksternal lainnya dengan sistem yang sedang dikembangkan. *Usecase* digunakan untuk melihat hubungan antara sistem dengan pengguna atau disebut juga sebagai aktor.

Pada *usecase* ini terdiri dari dua aktor yaitu Admin dan *user*, Admin sebagai operator melakukan enkripsi data sedangkan user menjadi penerima saja. Pada aplikasi ini admin dapat melakukan penulisan pesan, pengiriman pesan, dan enkripsi pesan sedangkan penerima pesan dapat menerima pesan, menyimpan pesan yang diterima atau mendownload setelah mendekripsinya





Gambar 3.
Usecase Diagram

IV. IMPLEMENTASI

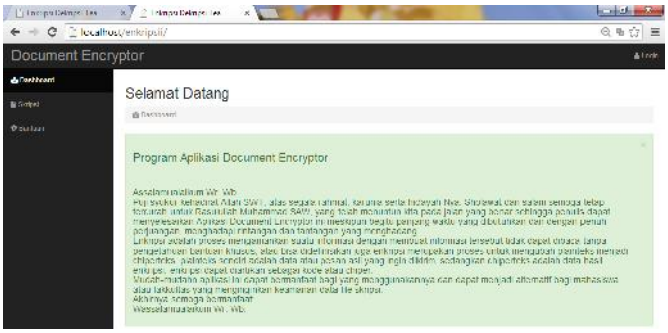
Berikut ini merupakan tampilan aplikasi yang penulis rancang yang terbagi dalam Halaman User Umum dan Halaman Administrator.

4.1 Halaman User Umum

Halaman user Umum adalah halaman yang diperuntukkan kepada user umum. dan bisa diakses oleh siapapun saja. Halaman user umum terdiri dari halaman Dashbor, Mahasiswa, Bantuan dan Hubungi Kami

4.1.1 Halaman Dashbor User Umum

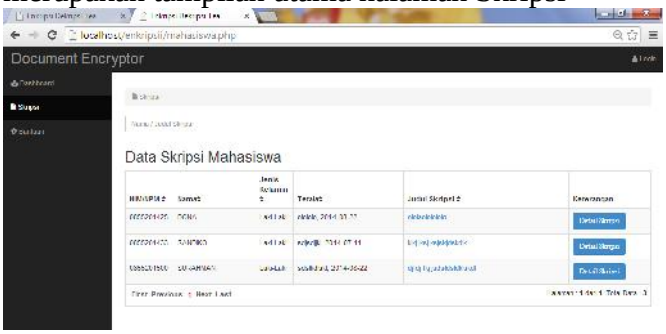
Halaman user Umum atau menu Dashbor merupakan tampilan yang muncul pada saat aplikasi pertamakali sedang dijalankan, Gambar 4 merupakan tampilan utama halaman User Umum



Gambar 4.
Tampilan Halaman User

4.1.2 Halaman Skripsi

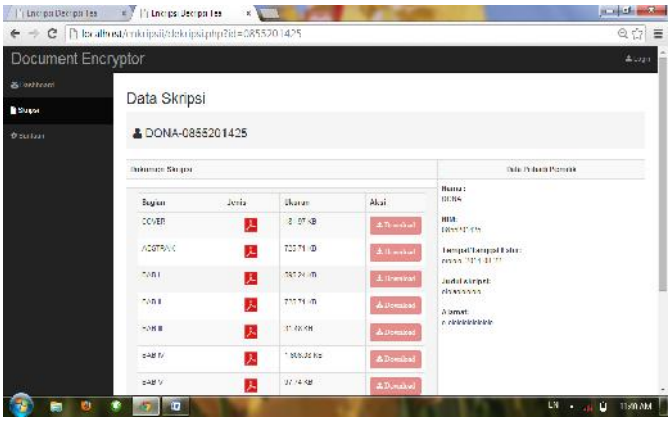
Halaman skripsi merupakan tampilan data skripsi yang sudah dienkripsi oleh admin, Gambar 5 merupakan tampilan utama halaman Skripsi



Gambar 5.
Tampilan Halaman Skripsi

4.1.3 Form Detail Skripsi

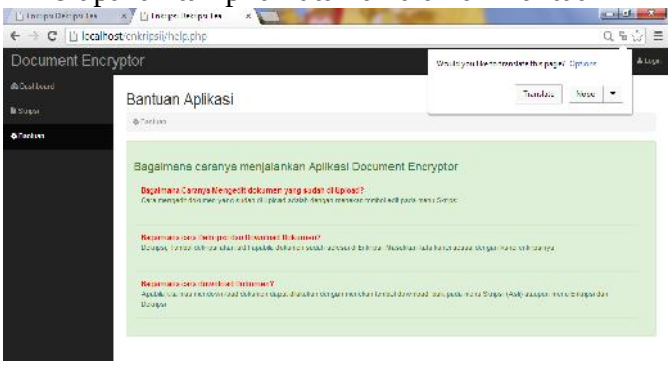
Form ini digunakan untuk melihat secara detail data skripsi, dan diform ini juga user dapat mendownload file skripsi dengan terlebih dahulu mengetahui kunci dekripsinya. Untuk melakukan dekripsi, klik tombol dekrip kemudian akan diminta kunci dekrip, masukkan kunci di form setelah kunci yang dimasukkan benar maka klik tombol download, jika menginginkan mendownload keseluruhan, dapat menggunakan tombol download semua dokumen



Gambar 6.
Tampilan Form Detail Skripsi

4.1.4 Halaman Bantuan

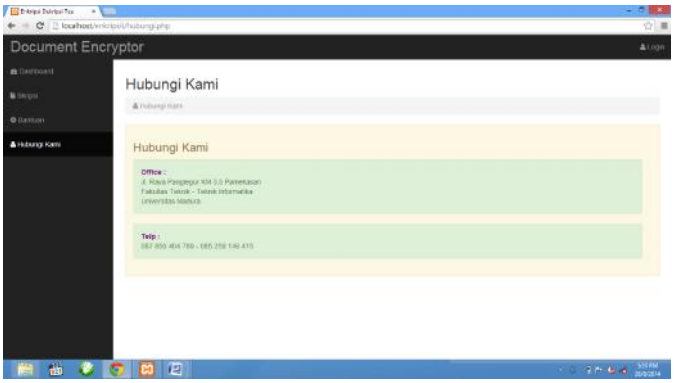
Halaman Bantuan merupakan tampilan pertanyaan dan jawaban yang dimaksudkan dapat membantu user jika ada ketidak pahaman, Gambar 7 merupakan tampilan utama halaman Bantuan



Gambar 7.
Tampilan Halaman Bantuan

4.1.5 Halaman Hubungi Kami

Halaman hubungi kami merupakan tampilan untuk informasi keberadaan administrator sehingga user dapat meminta kunci dekripsi dengan mudah. Berikut tampilan menu hubungi kami:



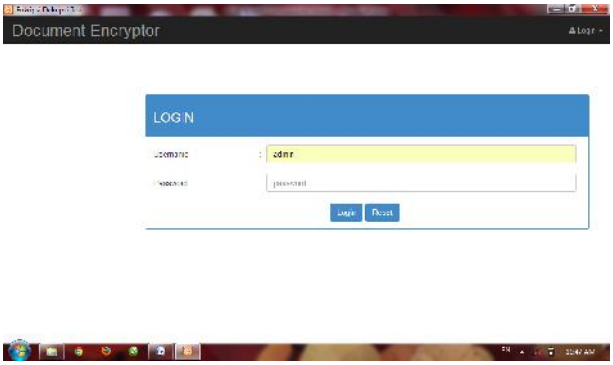
Gambar 8.
Tampilan Halaman Hubungi Kami

4.2 Halaman User Administrator

Halaman User Administrator merupakan halaman yang diperuntukkan kepada Administrator untuk mengolah Data. Halaman Administrator terdiri dari Halaman Login, Data Mahasiswa, Data Skripsi dan Bantuan.

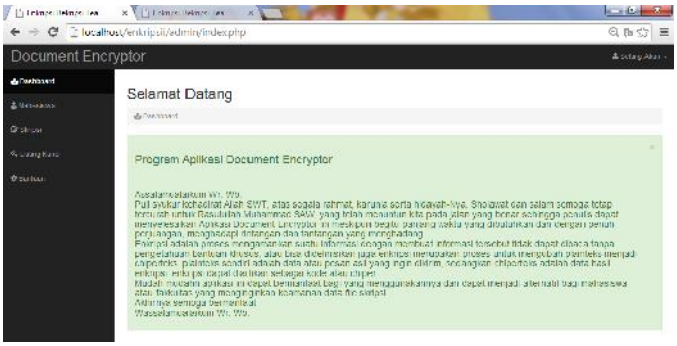
4.2.1 Tampilan Login

Tampilan *login* admin merupakan tampilan yang muncul pada saat aplikasi masuk pada halaman administrator. Dalam form ini terdapat menu *login.* Menu login ini difungsikan ketika seorang admin akan masuk kedalam program berikutnya. Gambar 4.6 merupakan tampilan *login* admin.



Gambar 9.
Tampilan *Login* Admin

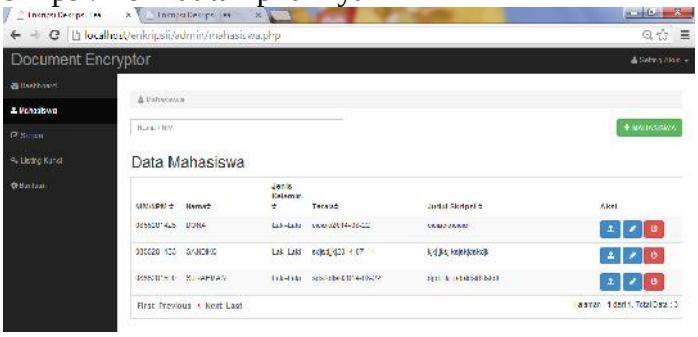
Setelah admin sukses login, maka akan masuk ke proses selanjutnya yaitu menu utama. Dimana pada *form* ini terdapat beberapa yaitu menu Dashbor, mahasiswa, menu skripsi, Menu Listing Kunci dan menu Bantuan.



Gambar 10.
Tampilan Dashbor Administrator

4.2.2 Menu Mahasiswa

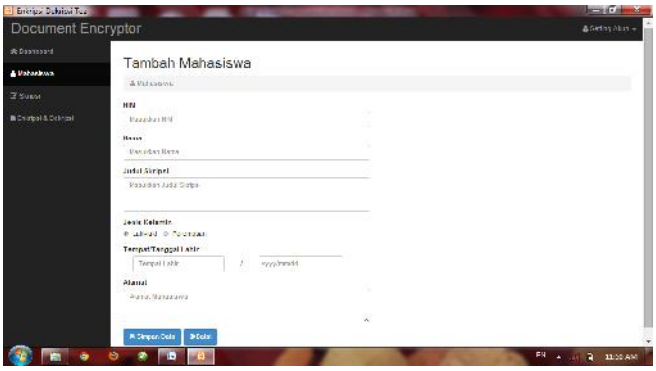
Pada menu mahasiswa ini terdapat data mahasiswa beberapa tombol perintah di antaranya tambah mahasiswa, ubah mahasiswa, hapus mahasiswa dan tombol unggah/*upload* dokumen skripsi. Berikut tampilannya



Gambar 11.
Tampilan Menu Mahasiswa

1. Form Tambah Mahasiswa

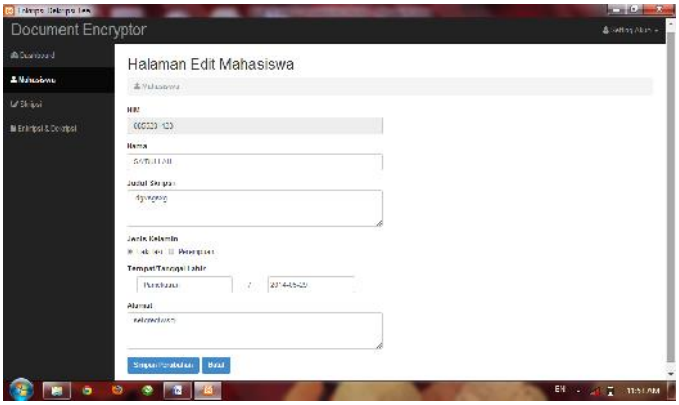
Form ini digunakan untuk menambah data mahasiswa yang sedang menempuh skripsi. Untuk menambah data mahasiswa klik tambah mahasiswa + MAHASISWA . Adapun data yang dibutuhkan adalah NIM, Nama, Judul Skripsi, Jenis Kelamin, Tempat/Tanggal lahir, dan alamat mahasiswa. setelah pengisian form selesai klik simpan data untuk menyimpan atau klik tombol batal apabila ingin membatalkan pengisian data mahasiswa.



Gambar 12.
Form Tambah Mahasiswa

2. Form edit mahasiswa

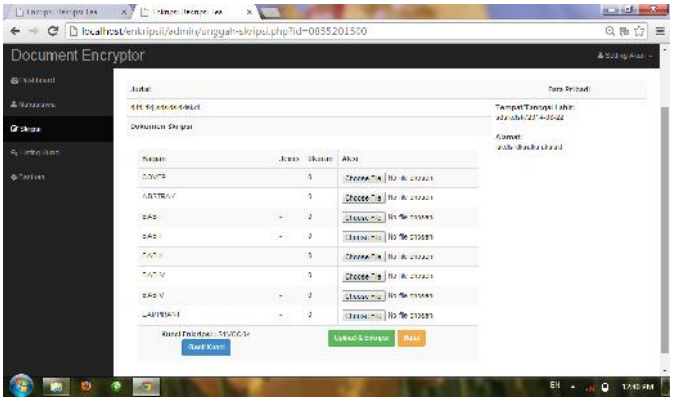
Form ini digunakan untuk mengedit data mahasiswa apabila terjadi kesalahan penginputan. Untuk melakukan perubahan silahkan klik tombol edit sehingga tampak pada gambar berikut. Kemudian klik simpan perubahan untuk menyimpan dan klik batal apabila ingin membatalkan penyimpanan.



Gambar 12.
Form Edit Mahasiswa

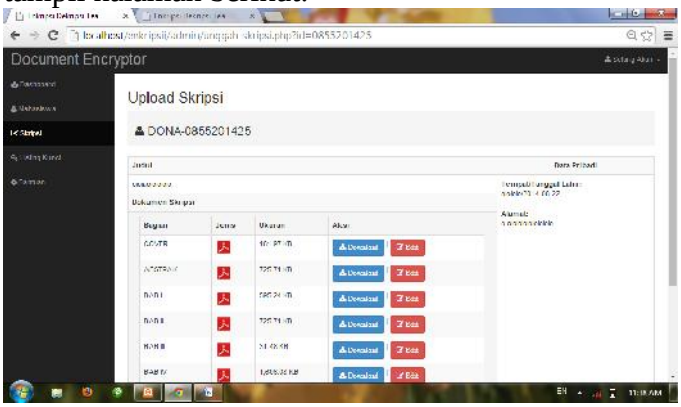
3. Form Upload skripsi

Form ini digunakan untuk mengunggah/mengUpload data skripsi mahasiswa. untuk mengunggah silahkan klik tombol unggah pada gambar di atas. Sehingga akan tampil seperti gambar berikut.



Gambar 13.
Form Upload Skripsi

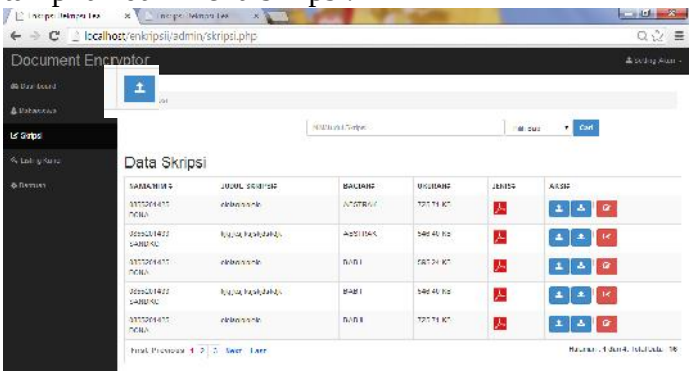
Setelah semua dokumen telah dipilih maka tekan tombol upload dan enkripsi maka program akan mengupload dan mengenkripsi data sehingga tampil halaman berikut:



Gambar 14.
Form download Skripsi

4.2.3 Menu Skripsi

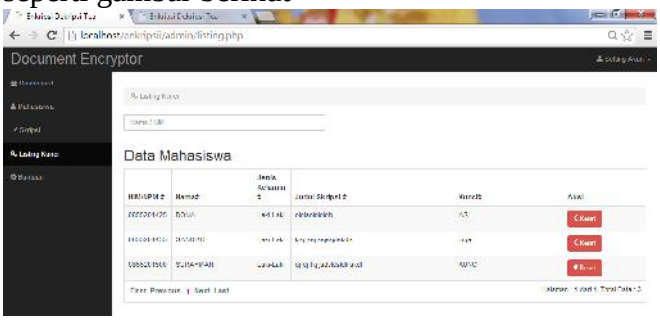
Pada menu ini terdapat beberapa fitur yaitu unggah skripsi, cari data skripsi, dan daftar data skripsi yang sudah tersimpan. Berikut merupakan tampilan dari menu skripsi.



Gambar 15.
Halaman Menu Skripsi

4.2.4 Menu Listing Kunci

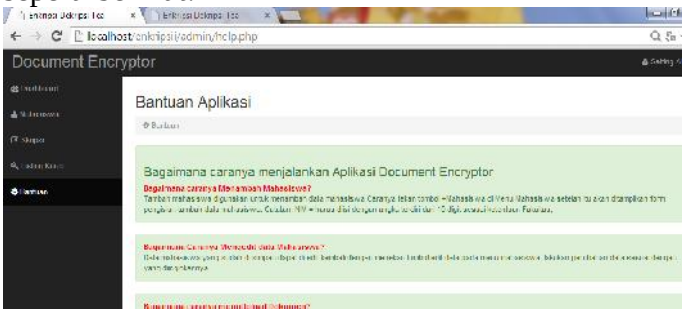
Halaman listing kunci dimaksudkan agar Admin tetap bias mengetahui kunci data skripsi yang sudah di enkripnya, dan pada menu ini juga dapat mereset password dengan menekan tombol reset, apabila dokumen sudah pernah didownload maka akan tampil keterangan downloaded sebagai informasi kepada administrator. sehimgga tampak seperti gambar berikut



Gambar 16.
Tampilan Menu Listing Kunci

4.2.5 Menu Bantuan

Menu bantuan dibuat sebagai menu yang dapat membantu dan memudahkan user menggunakan Aplikasi Doument Encryptor. Adapun tampilannya seperti berikut:



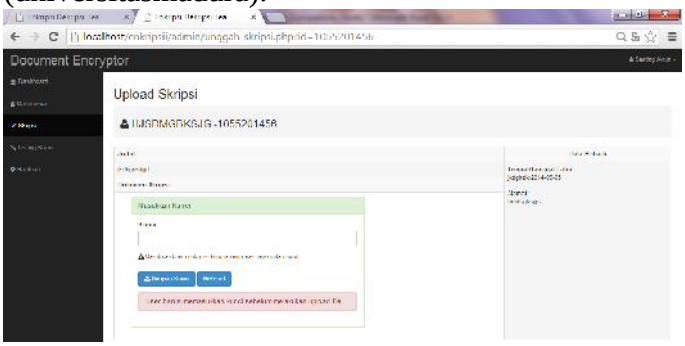
Gambar 17.
Tampilan Menu Bantuan

4.3 Hasil Pengujian pada Aplikasi

Pengujian pada Aplikasi dilakukan berdasarkan panjangnya kunci enkripsi dan dekripsi serta berdasarkan kapasitas file yang di enkripsi dan di dekripsi.

4.3.1 Pengujian File Skripsi Berdasarkan Panjangnya Kunci

Pengujian file skripsi berdasarkan panjangnya kunci merupakan pengujian dimana dilakukan Enkripsi dan Dekripsi pada file skripsi yang berbeda, file skripsi yang pertama menggunakan kunci yang pendek (as) dan yang kedua menggunakan kunci yang panjang (universitasmadura).



Gambar 18.
Pengujian Berdasarkan Panjang Kunci

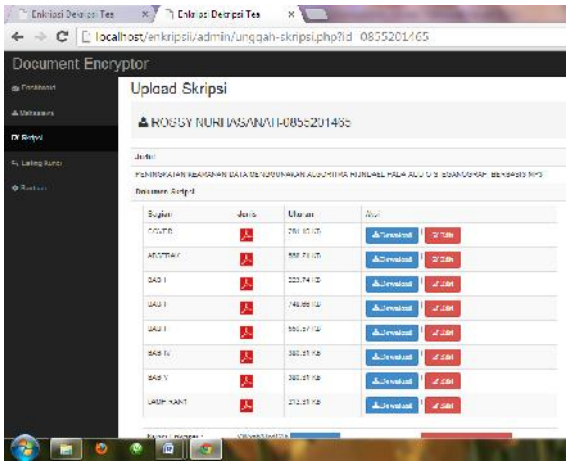
Dari hasil uji coba maka Panjang kunci tidak mempengaruhi Proses enkripsi Dekripsi dikarenakan *Tiny Encryption Algorithm (TEA)* memiliki 128 bit kunci yang disimpan dalam k[0], K[1], K[2], dan K[3], Apabila input kata kunci melebihi batas, maka akan dihapus secara otomatis.

Namun, penulis menyarankan penggunaan kunci yang agak panjang dan berkombinasi agar tidak mudah di pecahkan dan dapat menahan serangan kriptanalis.

4.3.2 Pengujian File Skripsi Berdasarkan Kapasitas File

Pengujian file skripsi berdasarkan kapasitas file, baik menggunakan Extension File yang berbeda dimana dilakukan Enkripsi pada file *.doc, dan *.pdf dengan kapasitas file yang berbeda.

Contoh pada gambar berikut peneliti menguji file skripsi *.pdf dengan *.doc



Gambar 19.
Pengujian Berdasarkan *Extension* File *.pdf dan.doc

Berdasarkan Uji coba yang dilakukan maka *extension* file yang berbeda tidak mempengaruhi Proses enkripsi dan dekripsi dokumen hanya saja kapasitas besarnya file yang di Enkripsi maupun Dekripsi akan banyak membutuhkan resource sehingga membutuhkan waktu yang sedikit lebih lama, dikarenakan Kapasitas File yang besar. Namun pada Proses Enkripsi dan Dekripsi data menggunakan *Algoritma Tiny Encryption Algorithm* (TEA), kapasitas plaintext dan chipertextnya sama sebelum atau sesudah di Enkripsi, artinya tidak ada perubahan Kapasitas File. Meskipun digunakan pada *extension* file yang berbeda.

V. KESIMPULAN

5.1 Kesimpulan

Setelah dilakukan pengujian terhadap Aplikasi dapat disimpulkan beberapa hal sebagai berikut:

1. Pada saat Proses Enkripsi dan Dekripsi dengan menggunakan kunci yang berbeda akan tetap dijadikan 128 bit. Sehingga jumlah kunci yang di Inputkan tidak berpengaruh pada lamanya Proses Enkripsi dan Dekripsi.
2. Algoritma kriptografi TEA merupakan algoritma yang aman digunakan hal ini dikarenakan jumlah round serta panjang kuncinya yang lebih panjang dan prosedur algoritma enkripsinya yang dirancang lebih kompleks, serta tidak membutuhkan S-Box dan P-Box dalam proses enkripsi dan dekripsinya sehingga meminimalkan penggunaan memory dan dapat memaksimalkan proses
3. File hasil enkripsi dan Dekripsi memiliki ukuran yang sama, Artinya metode *Tiny Encryption Algorithm* (TEA) ini tidak merubah ukuran file pada saat melakukan Enkripsi dan dekripsi, sehingga dapat dijadikan sebagai otentikasi data.

5.2 Saran

Berikut ini beberapa saran yang harus diperhatikan oleh pembaca:

1. Untuk pengembangan selanjutnya metode TEA ini bisa digunakan pada kasus lain seperti pada pengamanan file citra, audio dan video
2. Algoritma *Tiny Encryption Algorithm* (TEA) adalah algoritma kriptografi simetris, untuk pengembangan hendaknya menggunakan metode yang asimetris seperti : RSA dan ElGamal

REFERENSI

Berrent, Adam. (2012). *Advanced Encryption Standard by Example*, UK: ABI Software Development.

D. Wheeler and R.Needham (1994). *TEA a Tiny Encryption Algorithm*.

Heriyanto, Tedi. 1999. *Pengenalan Kriptografi*. Komputek: Jakarta.

Kromodimoeljo, Sentot. (2010). *Teori & Aplikasi Kriptografi*. SPK IT Consulting: Bandung.

Kurniawan, Yusuf. (2004) *Kriptografi Keamanan Internet dan Jaringan Komunikasi*, Informatika: Bandung.

Munir Rinaldi. (2006). *Kriptografi*. Informatika Bandung: Bandung

Purwanto, Edi, dkk. (2004). *Teknologi Informasi Dan Komunikasi*, Dirjen Dikti: Jakarta

Scott George, M. (2001). *Prinsip-Prinsip Sistem Informasi Manajemen*, Gramedia Pustaka: Jakarta

Halaman ini sengaja dikosongkan